ORNL/TM-2022/2659

Documentation on How to Run the NEAMS Workbench GUI on Sawtooth

Marc-Olivier G. Delchini
Robert A. Lefebvre

September 2022

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2022/2659

Nuclear Energy and Fuel Cycle Division

DOCUMENTATION ON HOW TO RUN THE NEAMS WORKBENCH GUI ON SAWTOOTH

Marc-Olivier G. Delchini
Robert Lefebvre

September 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-Battelle LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ORNL	Oak Ridge National Laboratory
INL	Idaho National Laboratory
NEAMS	Nuclear Energy Advanced Modeling and Simulation
HPC	high-performance computing
MOOSE	Multiphysics Object-Oriented Simulation Environment
NCRC	Nuclear Computational Resource Center
VTB	Virtual Test Bed
GUI	graphical user interface

ACKNOWLEDGMENTS

The authors would like to acknowledge Dr. Abdalla Abou Jaoude from Idaho National Laboratory and Dr. Emily Shemon from Argonne National Laboratory for reviewing the document and providing valuable comments.

ABSTRACT

This document provides guidelines on how to run MOOSE-based applications from the NEAMS Workbench GUI on the HPC platform Sawtooth located at Idaho National Laboratory. The different steps are illustrated with a multi-app example modeling a sodium-cooled fast reactor (SFR) taken from the virtual test bed (VTB) website. The workflow consists of four steps to demonstrate the capabilities of the NEAMS Workbench GUI, that covers log into Sawtooth, editing and validating input files, submission of the job to the queue and visualization of the numerical solution and the geometry. Each step is illustrated with figures in the text, and a demonstration video is also available.

1. INTRODUCTION

The mission of the US Department of Energy's Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program is to develop, apply, and deploy state-of-the-art predictive modeling and simulation tools for the design and analysis of current and future nuclear energy systems. NEAMS develops state-of-the-art scalable tools, such as the BISON fuel performance code [5], GRIFFIN reactor physics code [12], System Analysis Module (SAM) code [6], and Nek5000 computational fluid dynamics code [3]. NEAMS tools are also designed to be interoperable to construct customized multiphysics workflows. This flexibility is achieved by using the Multiphysics Object-Oriented Simulation Environment (MOOSE) framework [13], which is the foundation of several solvers and also provides the functionality to couple individual applications in loose or tight coupling schemes through the MOOSE MultiApp System [1].

The NEAMS Workbench [4, 14] initiative is designed to facilitate the transition from conventional design tools to more advanced modeling and simulation tools by providing a common user interface for model creation, review, execution, and visualization on personal computers and high-performance computing (HPC) platforms.

The MOOSE MultiApp System allows users to assemble individual codes together to create a customized multiphysics simulation for any reactor design. To couple codes together, a hierarchical workflow is invoked involving multiple input files linked together through input syntax which specifies execution order and data transfers between codes. Typically, the MOOSE-based super-application *BlueCRAB* [11] is invoked on a HPC platform. Examples of multiphysics problems specific to nuclear reactor applications are available in the Virtual Test Bed (VTB) repository [8].

Recent work has focused on integrating the NEAMS Workbench with both MOOSE-based tools as well as enabling job launch on INL HPC platforms (Sawtooth). This integration now permits users to easily run VTB examples (or their own models) using the *BlueCRAB* binary. Access to INL HPC and BlueCRAB is granted upon request through the Nuclear Computational Resource Center (NCRC), but the NEAMS Workbench is available to all INL HPC users automatically. This report aims to provide users with guidelines on how to run VTB examples with the NEAMS Workbench on INL HPC platforms. A demonstration video is also available to further illustrate all steps of the documentation, and it is referenced in the text when needed.

2. THE NEAMS WORKBENCH

The NEAMS Workbench [10, 9] is an open-source licensed GUI developed at Oak Ridge National Laboratory that integrates ParaView [2] for visualizing the numerical solution. In recent years, the NEAMS

Workbench has integrated MOOSE-based applications, among others. The NEAMS Workbench provides a user-friendly interface for creating, editing, and validating input files to be run on desktop computers or HPC platforms and for visualizing the numerical solution with the built-in ParaView visualization and analysis software. Its Python-based application run time environment, template engine, and domain-specific language processing tool enables the integration of a wide range of applications and advanced workflows [9].

3. CLONING THE VTB GITHUB REPOSITORY

Before running the NEAMS Workbench GUI on the INL HPC platform, it is recommended to clone the VTB GitHub repository to your home directory on Sawtooth to access the input files used in the subsequent sections. This can be achieved from the HPC OnDemand website by starting a Sawtooth terminal as shown in Figure 1.

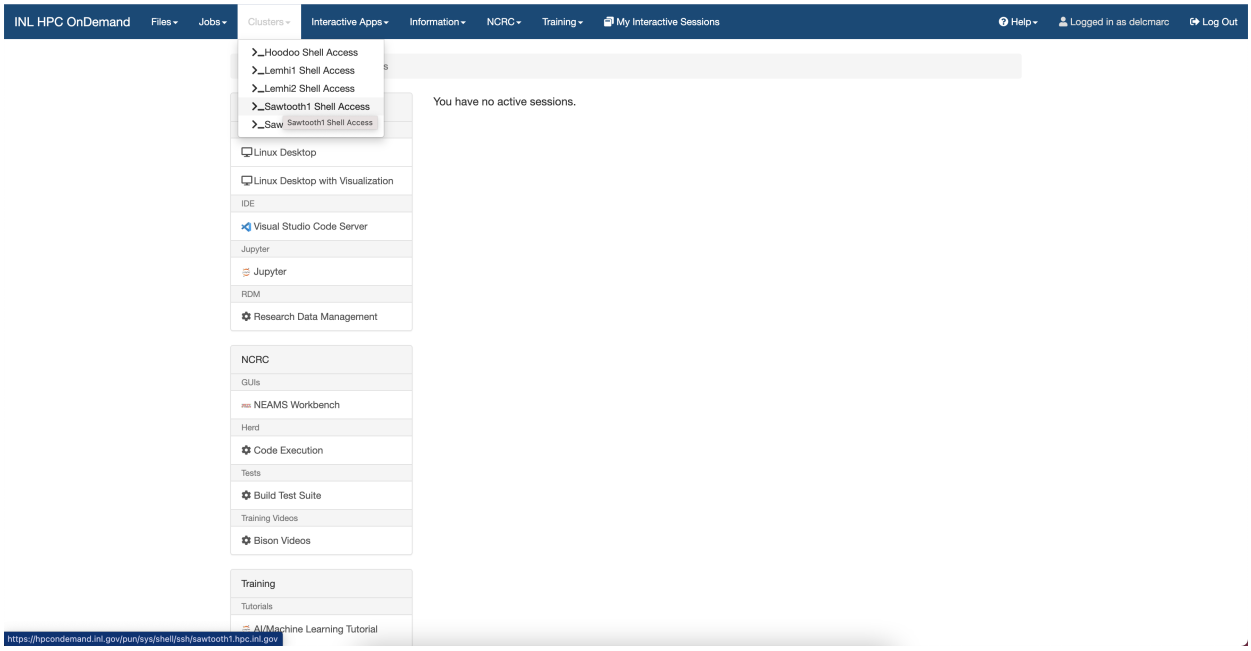

Figure 1. Accessing Sawtooth terminal on the Idaho National Laboratory HPC OnDemand website.

The Sawtooth terminal should immediately start as seen in Figure 2:

Once the terminal is opened, we are ready to clone VTB GitHub repository. The first step is to load the *gitlfs* module and to install *Git LFS* locally. Git-LFS is needed to clone large size files (mesh, database, ...) present in the VTB GitHub repository:

```
module load gitlfs; git lfs install
```

If successful, the terminal should show ‘Git LFS initialized.‘ Once git-lfs is enabled, copy the successive command lines to your Sawtooth terminal and hit enter to clone the VTB repository:

```
git clone https://github.com/idaholab/virtual_test_bed.git virtual_test_bed
```

The output of the above command line is as follows:

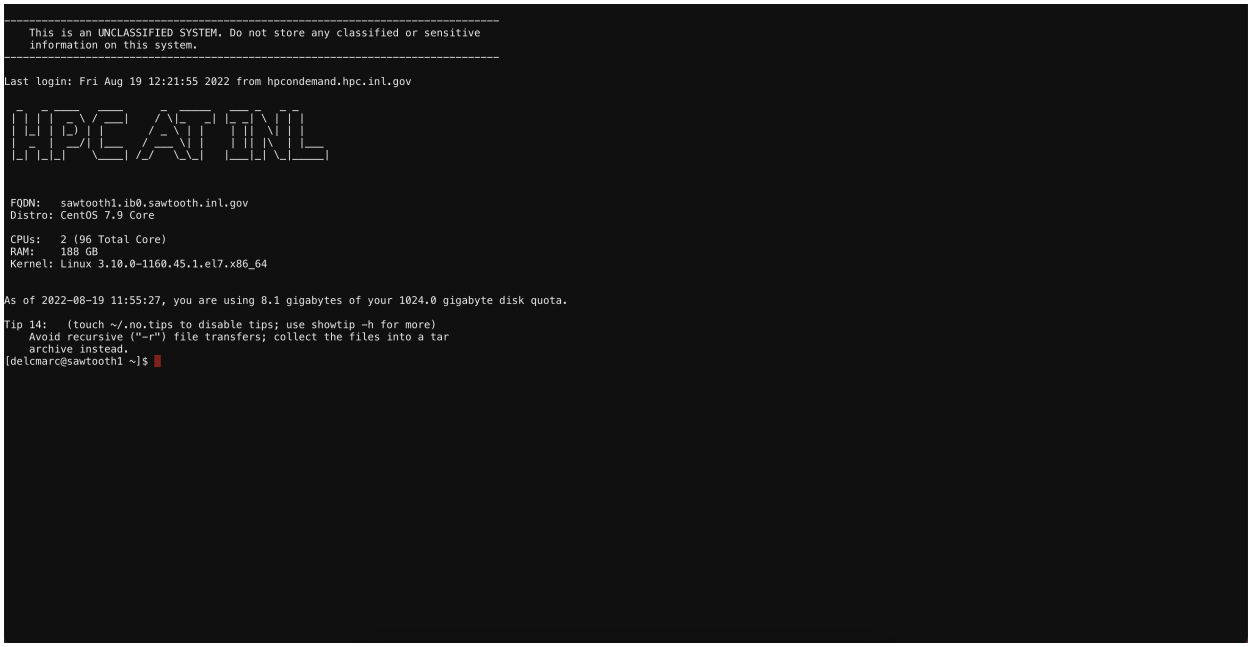

Figure 2. Sawtooth terminal on the Idaho National Laboratory HPC OnDemand website.

```
Cloning into 'virtual_test_bed'...
remote: Enumerating objects: 4524, done.
remote: Counting objects: 100% (885/885), done.
remote: Compressing objects: 100% (366/366), done.
remote: Total 4524 (delta 534), reused 719 (delta 502), pack-reused 3639
Receiving objects: 100% (4524/4524), 112.31 MiB | 39.01 MiB/s, done.
Resolving deltas: 100% (2487/2487), done.
Downloading htgr/htr10/data/mesh/htr-10-critical-a-rev6.e (2.7 MB)
Downloading htgr/htr10/data/mesh/htr-10-full-a-rev3.e (3.5 MB)
Downloading htgr/htr10/data/sph/htr-10-Diff-SPH.xml (3.0 MB)
Downloading htgr/htr10/data/xs/htr-10-XS.xml (4.9 MB)
Downloading htgr/htr10/steady/gold/out-HTR-10-full-Diff-SPH-ARO.e (7.4 MB)
Downloading htgr/htr10/steady/gold/out_HTR-10_critical.e (7.9 MB)
Downloading htgr/pbmr400/shared/oecd_pbmr400_tabulated_xs.xml (212 MB)
Downloading mrad/mesh/mrad_mesh.e (62 MB)
Downloading pbfhr/reflector/conduction/conduction.fld (247 MB)
Downloading pbfhr/reflector/meshes/fluid.re2 (60 MB)
Downloading pbfhr/reflector/meshes/solid.e (62 MB)
Checking out files: 100% (370/370), done.
```

The content of the *virtual_test_bed* directory can be accessed with the *ls* command:

```
ls virtual_test_bed/
apps COPYRIGHT doc htgr LICENSE mrad msr pbfhr README.md scripts
sfr
```

4. USING THE NEAMS WORKBENCH TO RUN A BLUECRAB MULTIPHYSICS MODEL ON SAWTOOTH

4.1 CONVENTIONAL WORKFLOW ON AN HPC PLATFORM

Conventionally, a modeling and simulation application is run on an HPC platform in a terminal by following four steps.

1. Connect the remote HPC platform using Secure Shell (https://www.ssh.com/academy/ssh) or one of its derivatives.
2. Enter the working directory of choice, then edit the input file and any other relevant files, including the mesh files.
3. Submit the job to the scheduler using a submission script from the working directory. The submission script contains logic to load the correct environment but also the executable to run and the number of cores to use.
4. Once the job is completed, visualize and analyze the numerical solution using a data visualization application, such as VisIt or ParaView. This may require copying large result files from the cluster to a desktop environment before data visualization can occur.

This workflow can be cumbersome to users unfamiliar with HPC platforms and an obstacle to potential industrial users in the nuclear engineering field who want to leverage HPC resources to speed up calculations. NCRC intends to increase the accessibility of modeling and simulation codes and the use of HPC resources by allowing users to access an HPC login node through a virtual desktop from the login page of the HPC OnDemand website (step 1). From there, users can run an application or open a terminal to edit files but still need their own submission script to submit jobs to the scheduler.

4.2 BRIDGING THE GAP WITH THE NEAMS WORKBENCH

The NEAMS Workbench is now available on the HPC OnDemand [7] website as a GUI under the *NCRC* tab and provides the means for performing steps 1–4. The following sections provide details on how to perform the above successive steps. All steps are also illustrated in a demonstration video.

4.2.1 Step 1: Creating a NEAMS Workbench Session on Sawtooth Using the HPC OnDemand Website

After connecting the HPC OnDemand website by entering the username and the passcode, the virtual Desktop opens as shown in Figure 3.

After clicking on the *NCRC*, select *NEAMS Workbench*, enter *Number of Hours* and *Cores* as shown in Figure 4, and click on *Launch* to submit the request to the scheduler. Note that the session will run the NEAMS Workbench GUI and not an actual job. Consequently, selecting one to four cores is sufficient. By default, the session runs on four cores with a wall time of 1 hour, which can be edited upon selection.

The session can take a few minutes to start Figure 5.

Once the session has started to run, the webpage is updated Figure 6 and the NEAMS Workbench GUI can be launched by clicking on *Launch NEAMS Workbench*. A virtual Linux Desktop opens and automatically starts the NEAMS Workbench GUI as displayed in Figure 7.

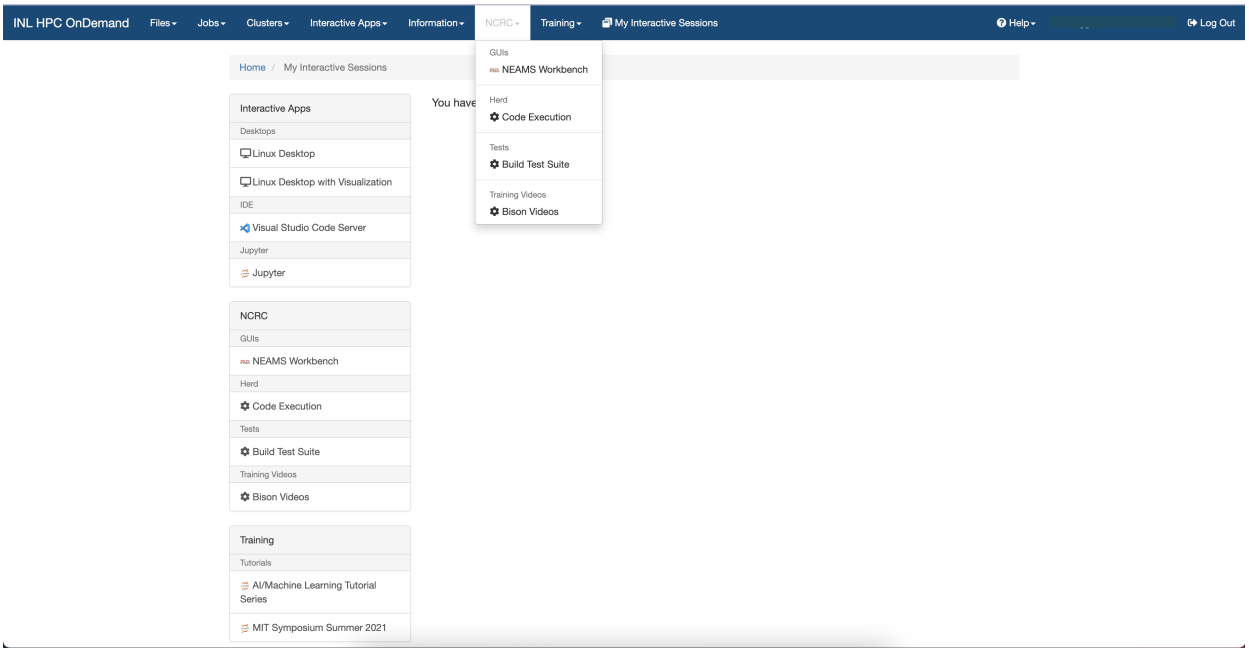

Figure 3. Accessing the NEAMS Workbench GUI on the Idaho National Laboratory HPC OnDemand website.

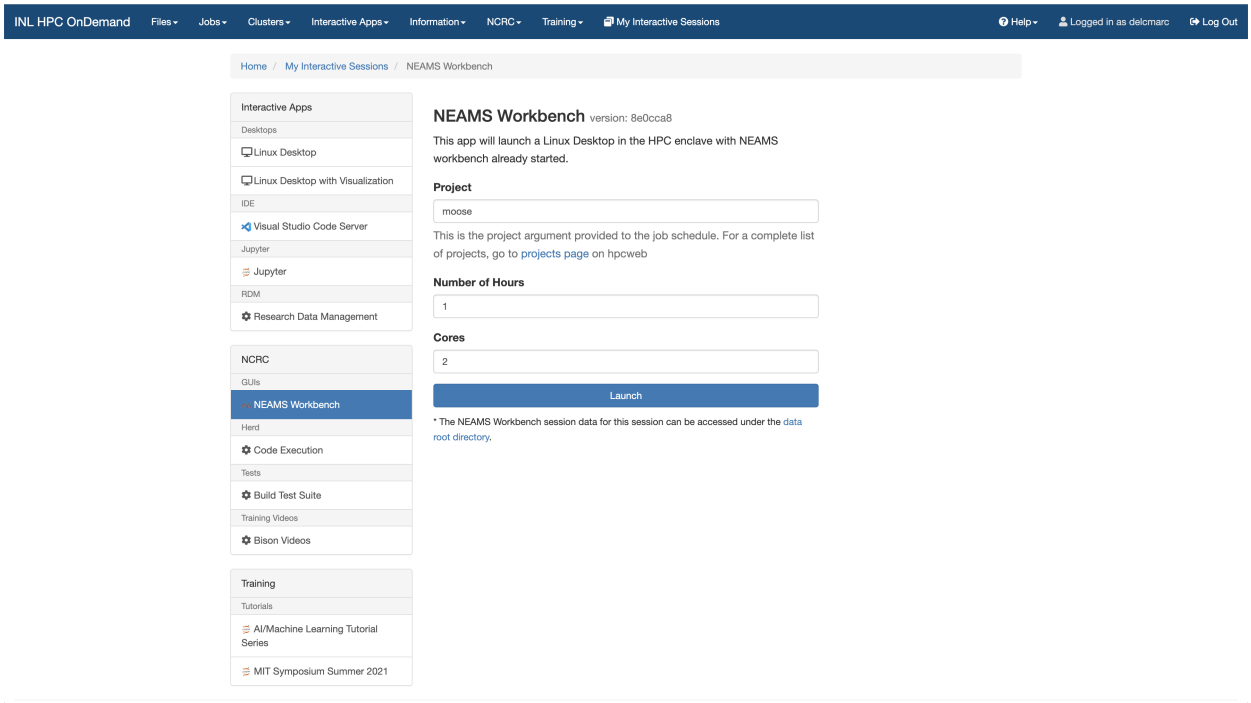

Figure 4. Create a NEAMS Workbench session on Sawtooth.

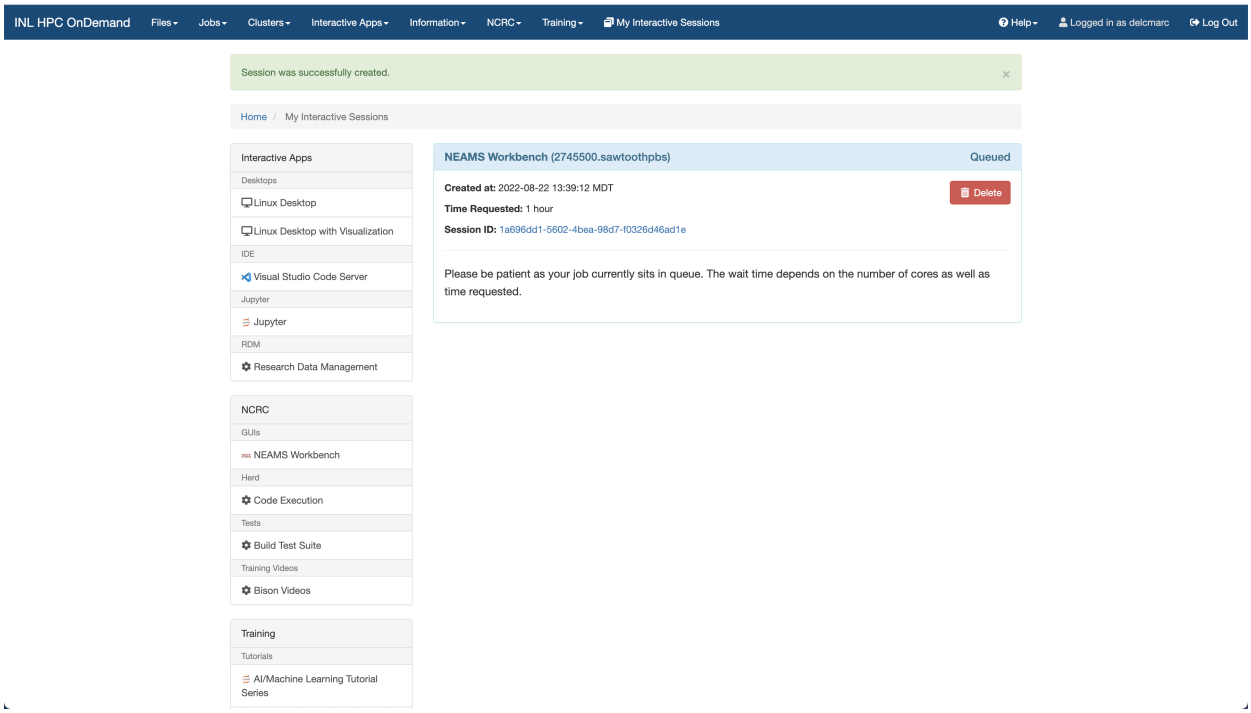

Figure 5. Waiting for a NEAMS Workbench session to start.

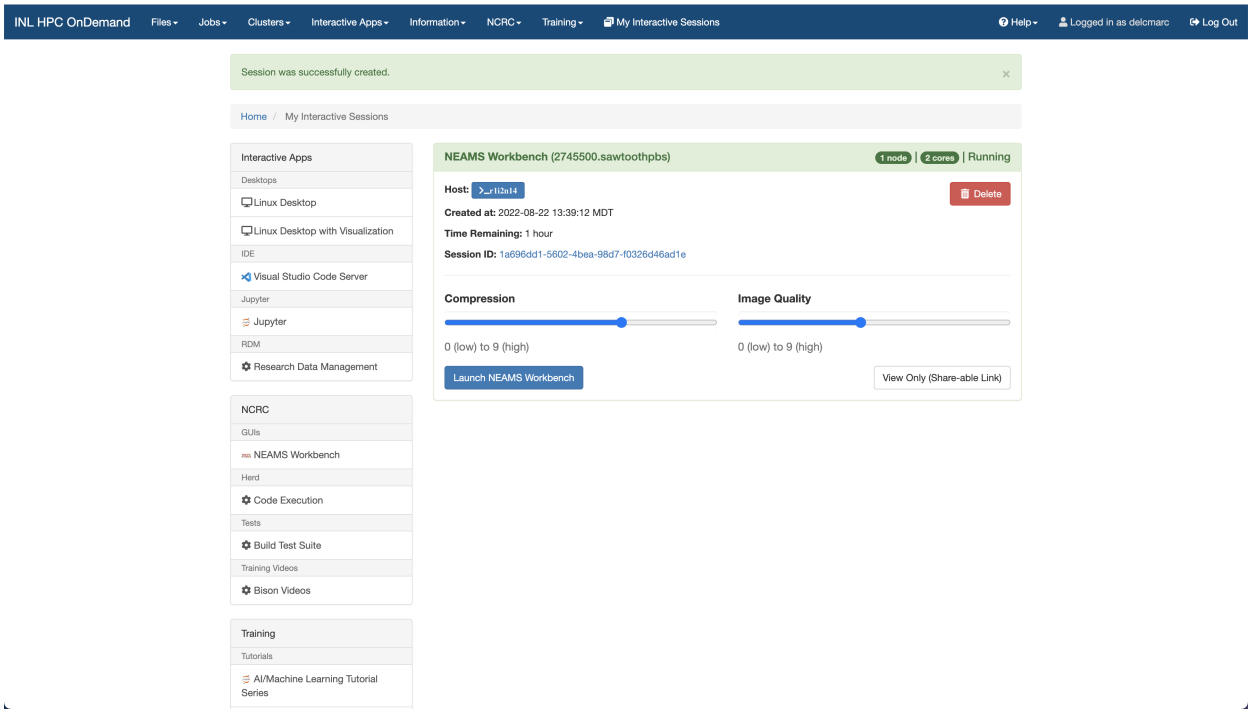

Figure 6. Launch NEAMS Workbench session on Sawtooth.

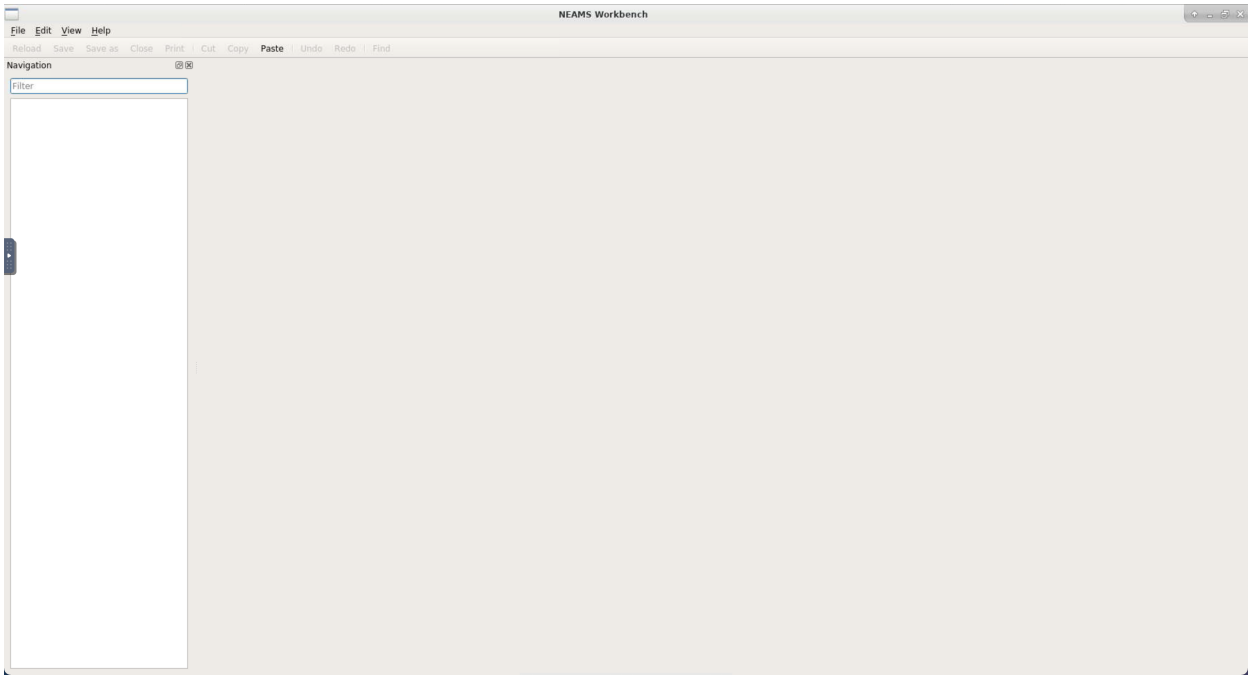

Figure 7. NEAMS Workbench GUI on Sawtooth.

4.2.2 Step 2: Enabling the BlueCrab Executable and Uploading an Input File to Edit

Once the NEAMS Workbench GUI (see step 1) has been initialized the user should set the environment for the *Blue-Crab* application which consists of loading modules and making the executable available in the path. This is achieved through the NEAMS Workbench GUI by clicking *File* and *Localhost*. A widget opens and list all MOOSE applications available on Sawtooth. The widget should be enlarged with the mouse cursor. The *Blue Crab* application or any other MOOSE-based applications to activate is selected by (1) clicking on the number left to the application name and (2) *Activate*. The output of the process is shown in the tab below the list of all MOOSE-based application and is deemed successful once the content *{"Completed" : true}* is displayed as illustrated in Figure 8. Note that activation process sets up the correct Linux environment for the MOOSE-based application and also loads various files needed by the NEAMS Workbench GUI for auto-completion and validations of the input files.

The next step consists of opening an input file from the NEAMS Workbench GUI, which is demonstrated with the sodium-cooled fast reactor example (SFR) from the VTB website [8]. Assuming that the VTB repository files were already cloned to the INL HPC filesystem (see Section 3.), several input files are available and can be navigated to within Workbench. Users can load the *griffin.i* input files (main application or main app) for the SFR VTB example by clicking *File* and then *Open File*. A window should open and let you navigate through the content of the home directory. Click on *virtual_test_bed* and then *sfr* to access the input files Figure 9.

After opening the input file, the navigation tab shows all input file blocks, sub-blocks, and their content to ease navigation in the input file. Each block can be extended to show its content. Make sure the NEAMS Workbench automatically updates the correct schema and runtime environment to ncrcbluecrab - ncrcbluecrab (left and right of the dropdown box named document). The Exodus mesh file

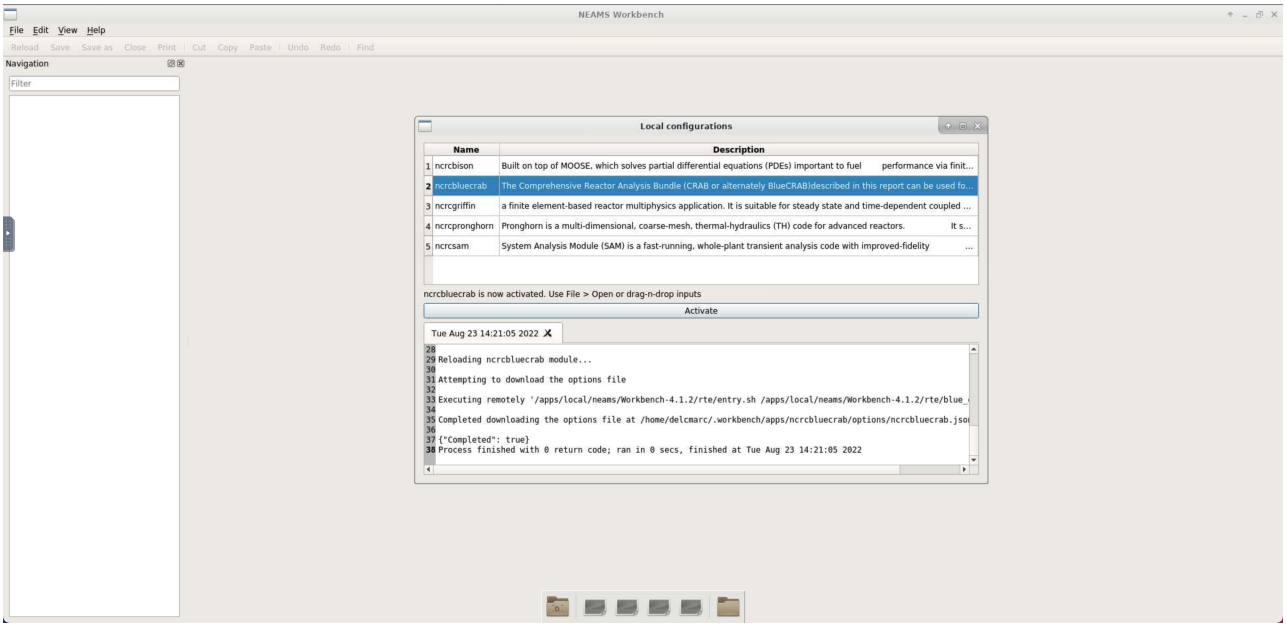

Figure 8. Enabling the BlueCrab environment from the NEAMS Workbench GUI.

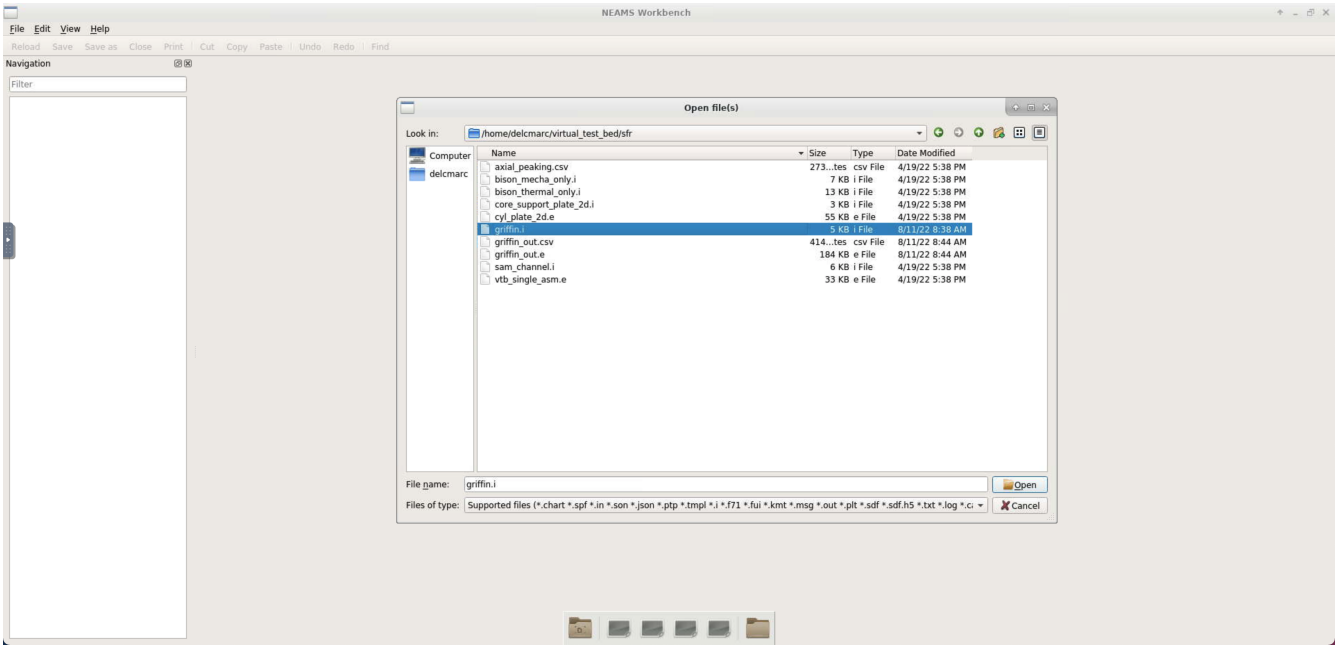

Figure 9. Open *griffin.i* input file with the NEAMS Workbench.

vtb_single_asm.e can also be opened within the NEAMS Workbench by following the same steps for the main app input file. Once opened and after splitting screen (click on the tab name *srf:vtb_single_asme.e* and drag and drop in the right-hand side of the GUI), users can highlight specific mesh blocks from the input file by right-clicking on the mesh block name and selecting *Inspect*, as shown for mesh block 2 (line 67) in Figure 10. Users can also easily identify geometry components to assign material properties, kernels, initial conditions, or boundary conditions. The same capabilities are also available for all sub-applications (sub-app) input files.

Validation of the input file is automatically handled by the NEAMS Workbench GUI and validation errors are shown in the *Validation* tab highlighted in red and located at the bottom right of the GUI (see Figure 10). It should be noted that some of the validation errors are due to some syntax not being supported by the NEAMS Workbench. These issues are being addressed and should be fixed in future release of the NEAMS Workbench GUI. By clicking on the validation error message, the user is taken to the line with the syntax error. Auto-completion is also available to help with file editing which is illustrated at time 1 : 41 in the demonstration video.

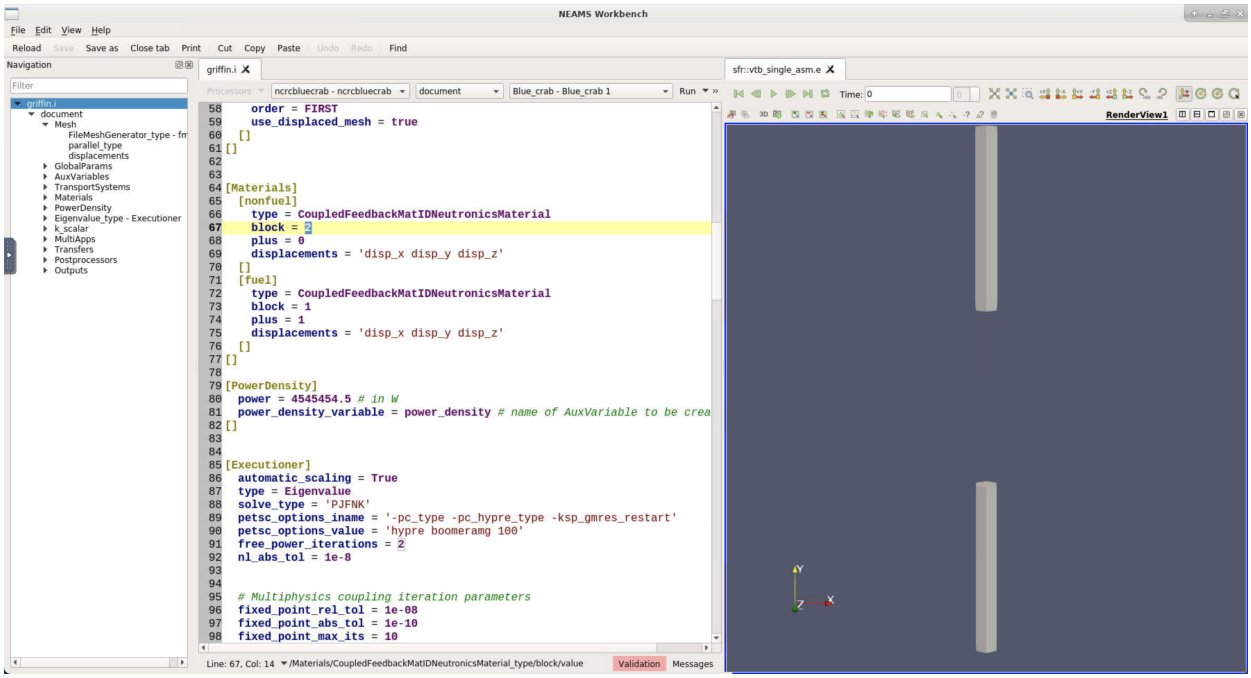

Figure 10. Input file (left) and mesh visualization (right) with the NEAMS Workbench.

4.2.3 Step 3: Running a Job From the NEAMS Workbench GUI

After editing the input file and making sure that the physics enabled is consistent with the geometry visualized with ParaView, the input file is submitted to the scheduler. A portable bash script (PBS) scheduler is available on Sawtooth and PBS directives can be updated if needed (number of nodes, cores, wall time, etc ..) or left to their default values (one node, one CPU and a wall time of ten minutes). To edit the PBS directive, click on the back downward arrow right to the *Run* button and select *Customize Run Options...*. A window should open to show the PBS directives after scrolling down. Edit the entries to your convenience, click *Apply* and then *Ok* to close the widget as shown in Figure 11. To submit the job to the

queue managed by the PBS scheduler, click *Run*. The *Message* tab should display the output from the run while the NEAMS Workbench checks on the status of the job.

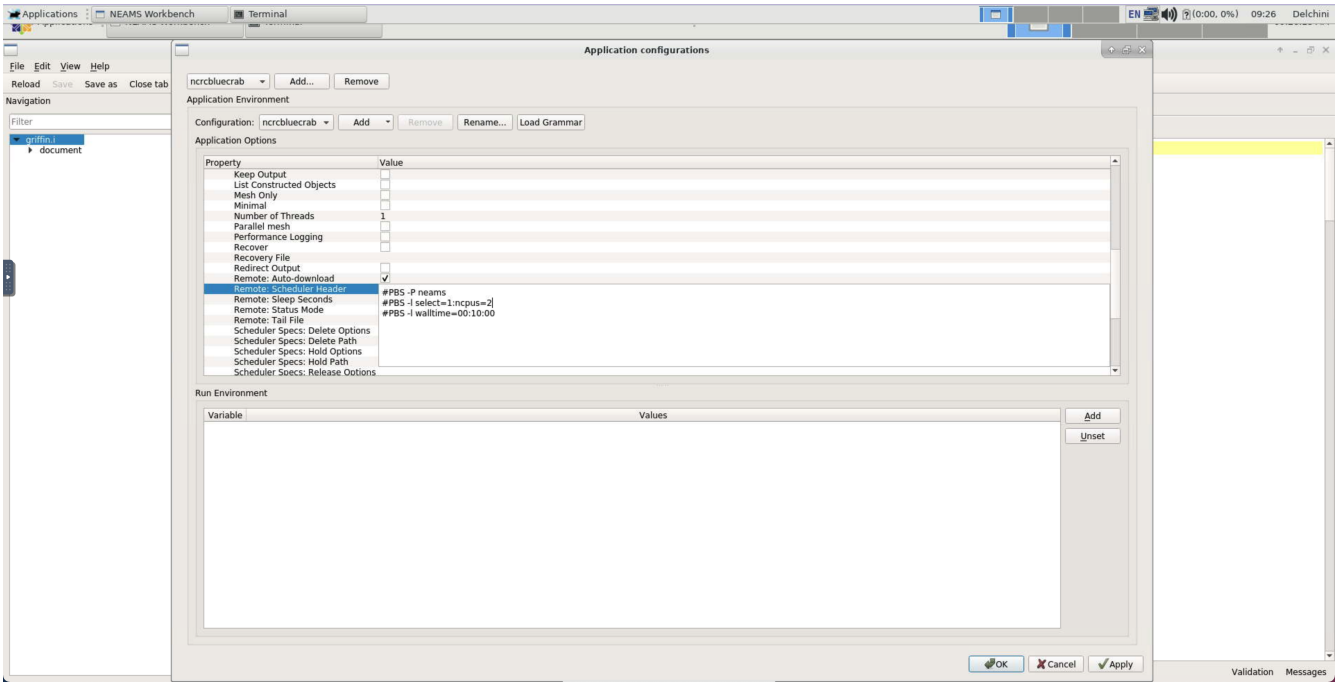

Figure 11. Editing PBS options with the NEAMS Workbench GUI.

4.2.4 Step 4: Visualize the Numerical Solution From the NEAMS Workbench GUI

Visualization of the numerical solution is also achieved within the NEAMS Workbench GUI. Users can refer to the steps below or watch the demonstration video starting at 4:20. As for visualizing the geometry and the mesh described in Step 1, the Exodus file that contains the numerical solution can be visualized with the built-in ParaView, as shown in Figure 16. The same steps as in Section 4.2.1 can be followed. An alternative is to right click on the name of the input file in the *Navigation* tab and select *Open working directory*. From there, a navigation window opens with all files contained in the working directory. The Exodus file *griffin_out.e* can be dragged and dropped in the NEAMS Workbench GUI. A *visualization tab* or *render view* should automatically open and display the geometry as shown in Figure 12.

The ParaView GUI is enabled by clicking on *Visualization* and checking the box for *Visualization GUI*. The *Visualization GUI* appears in the right-hand side of the NEAMS Workbench GUI with ParaView options)Figure 13). After enabling all variables and clicking *Apply* within the *Visualization GUI*, a variable can be plotted by right-clicking **on the geometry** in the *visualization tab* and selecting *Color by* and then the name of the variable of interest (Figure 14 and Figure 15). The *Visualization tab* can also be split to plot multiple variables side-by-side, as seen in Figure 16.

5. CONSIDERATIONS

The scheduler options (e.g., wall time, number of nodes and cores) can be modified within the NEAMS Workbench before submission to Sawtooth by clicking on the the downward black arrow to the right of the

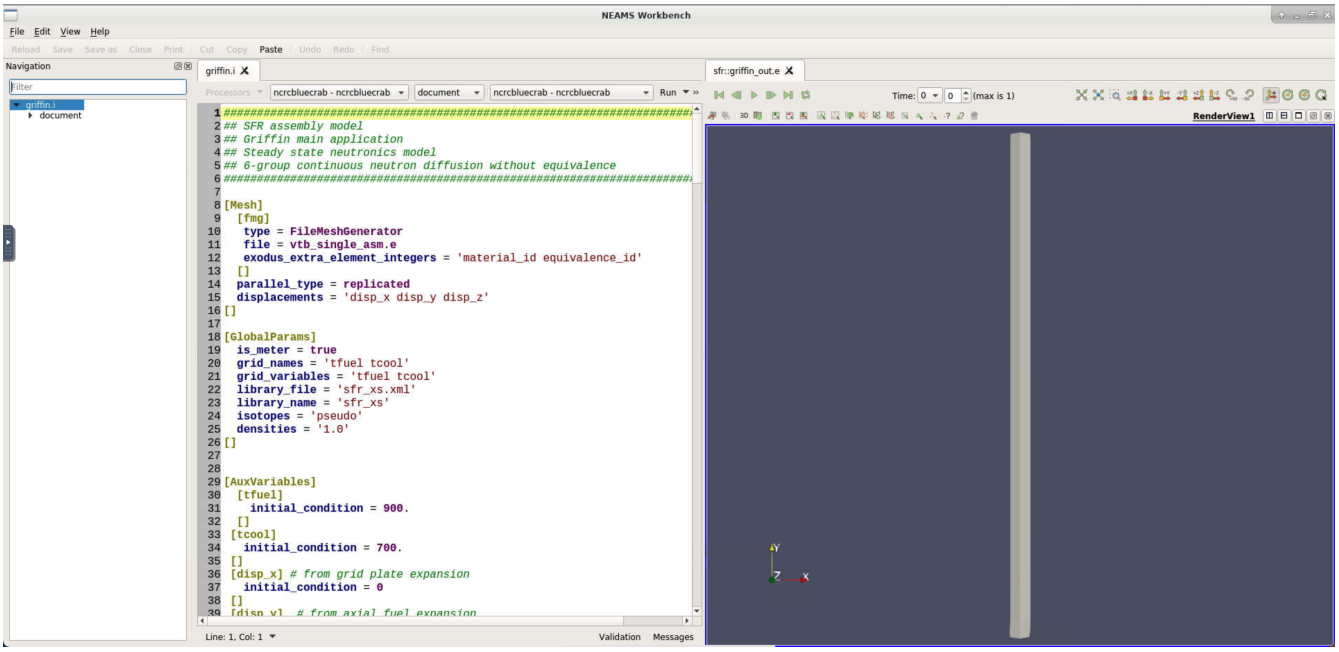

Figure 12. Drag and drop the Exodus solution output file *griffin_out.e* in the NEAMS Workbench GUI to open a visualization tab or render view.

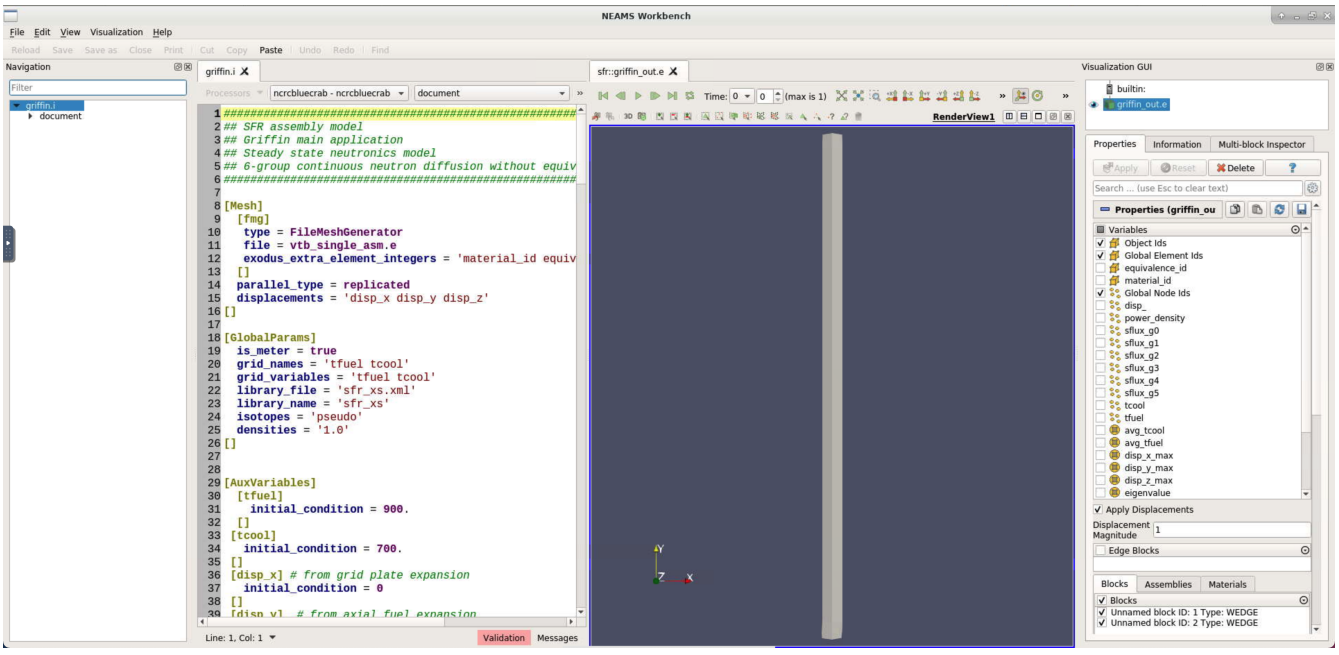

Figure 13. Enabling the Visualization GUI for ParaView.

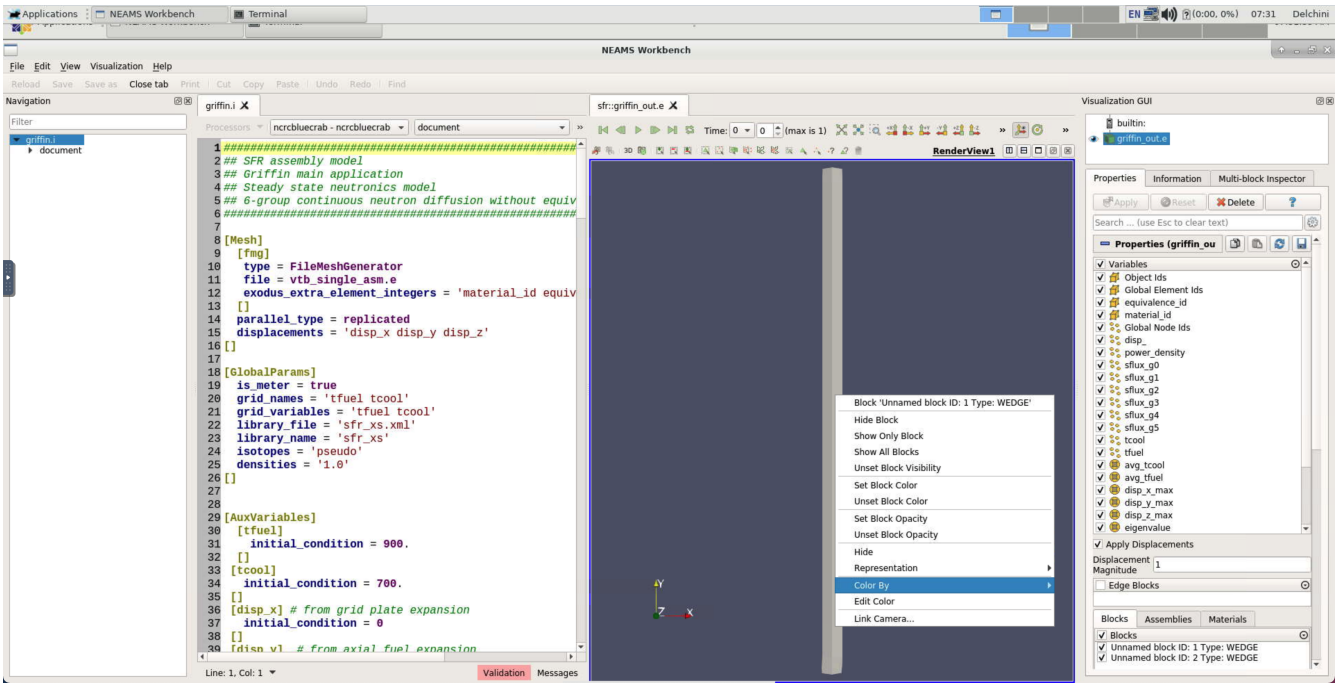

Figure 14. After right-clicking on the geometry in the *Visualization tab*, select 'Color by' to access all variables to plot.

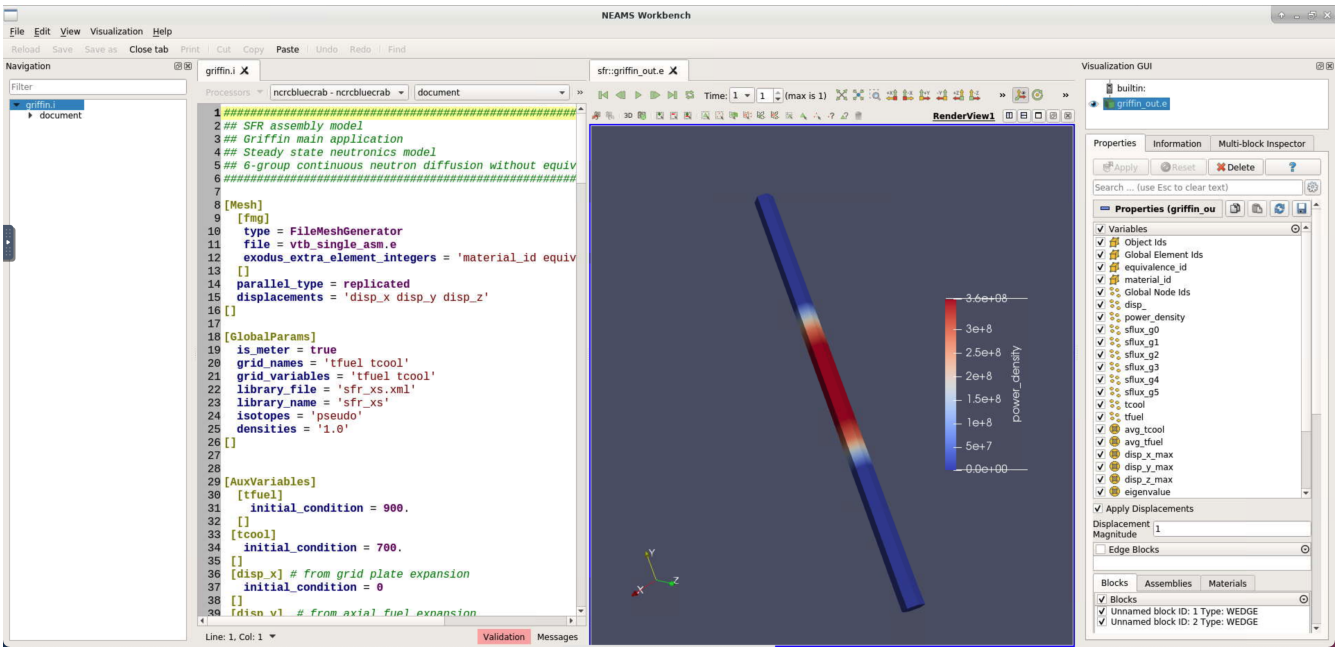

Figure 15. NEAMS Workbench GUI showing *power density* profile in the *Visualization tab*.

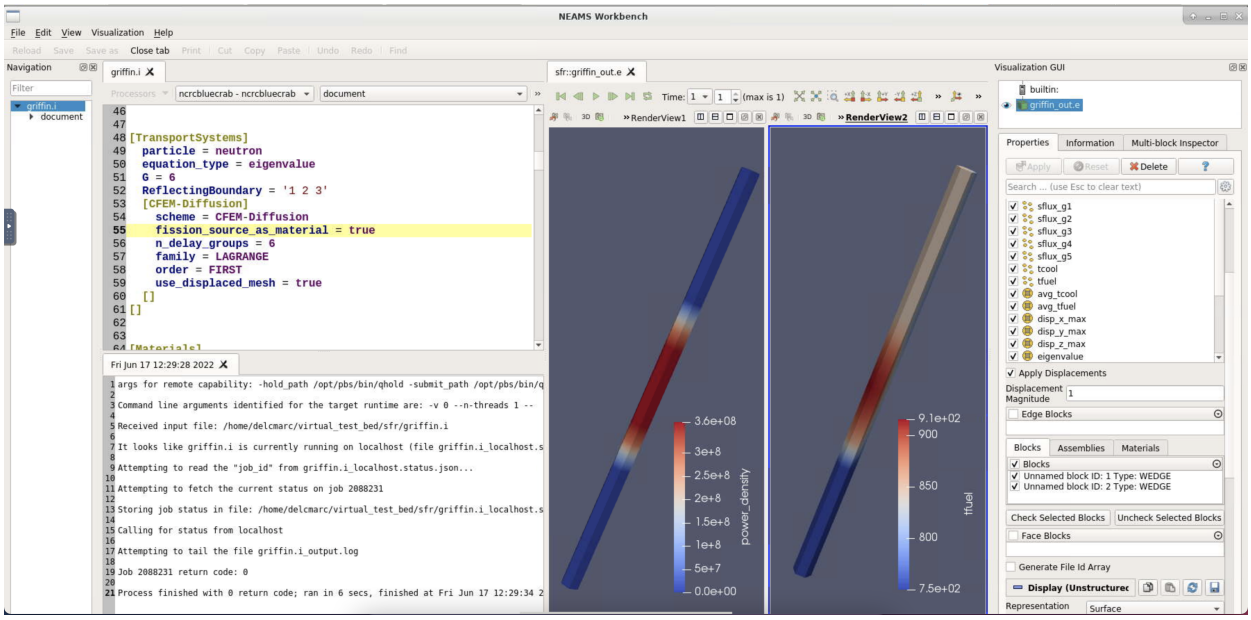

Figure 16. The NEAMS Workbench GUI with *griffin.i* input file (top left), the output from the scheduler (bottom left), and the numerical solution visualized with ParaView (right).

Run button and selecting *Customize Run Options*. Users can then edit the default entries with values of their choice.

Potential users should remember that this workflow is still under active development and that some VTB examples are not fully supported. For instance, the molten salt fast reactor VTB example relies on Nek5000-v19 to inform the Pronghorn model. Version 19 of Nek5000 execution is not currently integrated with the NEAMS Workbench; only version 17 execution is supported.

Users may also see validation error messages when opening some of the VTB input files with the NEAMS Workbench, as shown at the bottom of the *griffin.i* input file in Figure 9. This is expected because some of the advanced MOOSE syntax requires a language server protocol to be fully cross-referenced and validated across multiple input blocks and input files. This work is in progress and is discussed briefly in the following section.

6. CONCLUSIONS AND FUTURE WORK

This paper presents the latest workflow for running a VTB multiphysics model with the NEAMS Workbench on the INL HPC platform. Workbench is immediately and automatically available to INL HPC users, and its integration with BlueCRAB and other MOOSE-based NEAMS tools is automatically mapped based on user license permissions. Together, the HPC OnDemand website and the NEAMS Workbench GUI enhance user experience by automating most of the tasks needed to run a job on an HPC platform and significantly reducing the barrier to using NEAMS tools. The NEAMS Workbench provides a unique GUI to all MOOSE-based NEAMS tools and allows new users to quickly run on the INL HPC platform while focusing on developing the physical models and analyzing the numerical solutions. In the long term, this improved graphical tool will likely be the cornerstone for deploying NEAMS tools and the workflow used

in the nuclear industry to develop and design new advanced nuclear reactor concepts.

As future work, a language server protocol will be added to the NEAMS Workbench to enable enhanced validation of input parameters across multiple input files and real-time generation of the validation logic from the executable. Further work will also be performed to integrate Nek5000-v19 [3], which is used in the molten salt reactor model, and demonstrate the use of Dakota for sensitivity analyses with multiphysics problems on the INL HPC platforms.

References

[1] A. Abou-Jaoude, D. Gaston, G. Giudicelli, B. Feng, and C. Permann. The virtual test bed repository : A library of multiphysics reference reactor models using neams tools. In *Transactions of the American Nuclear Society*, 2021.

[2] James Ahrens, Sébastien Jourdain, P. O'Leary, John Patchett, David H Rogers, and Mark Petersen. Proceedings of the international conference for high performance computing. In *Proceedings of the International Conference for High Performance Computing*, 2014.

[3] Argonne National Laboratory. Nek5000. https://github.com/Nek5000/Nek5000, June 2022.

[4] Marc-Olivier G. Delchini, Laura P. Swiler, and Robert A. Lefebvre. Extension of the neams workbench to parallel sensitivity and uncertainty analysis of thermal hydraulic parameters using dakota and nek5000. *Nuclear Engineering and Technology*, 53(10):3449–3459, 2021.

[5] D Gaston, C Newman, G Hansen, and D Lebrun-Grandie. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nucl. Eng. Design*, 239:1768–1778, 2009.

[6] Rui Hu, Ling Zou, and Guojun Hu. Sam user's guide. 8 2019.

[7] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf, and Brian L. McMichael. Open ondemand: A web-based client portal for hpc centers. *Journal of Open Source Software*, 3(25):622, 2018.

[8] Idaho National Laboratory. Virtual Test Bed. https://mooseframework.inl.gov/virtual_test_bed/, June 2022.

[9] R. A. Lefebvre, B. R. Langley, P. Miller, M. Delchini, M. L. Baird, and J. P. Lefebvre. NEAMS Workbench Status and Capabilities. Technical Report ORNL/TM-2019/1314, Oak Ridge National Laboratory, 2019.

[10] R. A. Lefebvre, B. R. Langley, and A. B. Thompson. Report on NEAMS Workbench Support for MOOSE Applications. Technical Report M3MS-16OR0401086, Oak Ridge National Laboratory, 2016.

[11] R. Martineau, D. Andrs, R. Carlsen, D. Gaston, J. Hansel, F. Kong, A. Lindsay, C. Permann, A. Slaughter, E. Merzari, Rui Hu, A. Novak, and R. Slaybaugh. Multiphysics for nuclear energy applications using a cohesive computational framework. *Nuclear Engineering and Design*, 367:110751, 2020.

[12] Javier Ortensi, Yaqi Wang, Vincent M Laboure, Zachary M Prince, Changho Lee, Yeon Sang Jung, Hansol Park, and Emily R. Shemon. Griffin software development plan. 6 2021.

[13] Cody J. Permann, Derek R. Gaston, David Andrš, Robert W. Carlsen, Fande Kong, Alexander D. Lindsay, Jason M. Miller, John W. Peterson, Andrew E. Slaughter, Roy H. Stogner, and Richard C. Martineau. MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11:100430, 2020.

[14] N. Stauff, P. Lartaud, Y. S. Jung, C. H. Lee, K Zeng, and Jason Hou. Status of the neams and arc neutronic fast reactor tools integration to the neams workbench. 9 2019.